

Relatório Implementação Automato Finito Determinístico

Felipe Helfensteler Beskow

2019/01

Para a implementação do Automato Finito Determinístico foi escolhida a linguagem Python por conta de sua facilidade e compacidade no código. Havia escolhido inicialmente a linguagem C, porém por conta da complexidade de lidar com ponteiro, já que idealizei usar uma lista encadeada acabei desistindo.

O código em Python ficou da seguinte forma:

```
from sys import argv

arq = []
transicao = []

def leArquivo(localArq):
    arquivo = open(localArq, "r")
    for linha in arquivo.readlines():
        arq.append(linha[:-1])
    arquivo.close()

def separaArquivo():
    x=3
    while x<len(arq):
        transicao.append(arq[x].split(" "))
        x+=1

def defineTransicao(est):
    if est == "":
        return -1
    else:
        return int(est)

def automato():
    k = 0
    estado = 0
    palavra = arq[0]
    while k < len(palavra):
        est=estado
        if palavra[k] == '0':
            estado = defineTransicao(transicao[estado][1][1:])
```

```

        else:
            estado = defineTransicao(transicao[estado][2][1:])
        if estado == -1:
            break
        print("Delta(q{},{})=q{}".format(est,palavra[k],estado))
        k+=1
    if estado == int(arq[2][1:]):
        print("A cadeia foi aceita")
    else:
        print("A cadeia foi rejeitada")

leArquivo(argv[1])

separaArquivo()

automato()

```

Inicialmente no programa importei o módulo que será usado para obter os argumentos passados ao programa:

```
from sys import argv
```

Após isso declarei as listas que iremos usar. Por mais que Python seja uma linguagem que não precise que você declare as variáveis, fiz isso para que pudesse dar o comando “append” mais a frente.

```

arq = []
transicao = []

```

Criei a função “leArquivo” para ler somente o arquivo e armazenar os valores do arquivo que estamos lendo. As informações ficarão salvas na lista “arq”.

```

def leArquivo(localArq):
    arquivo = open(localArq, "r")
    for linha in arquivo.readlines():
        arq.append(linha[:-1])
    arquivo.close()

```

A função “separaArquivo” pega a lista “arq” e separa a parte das transições em uma lista a parte chamada “transicao”. A função split separa a string lida em uma lista que é adicionada a lista “transicao” criando uma lista de listas.

```

def separaArquivo():
    x=3
    while x<len(arq):
        transicao.append(arq[x].split(" "))
        x+=1

```

A função “automato” é a cereja do bolo do código. Aqui está a lógica central do programa. Após setar as variáveis, o *while* percorrerá a palavra que iremos testar. A cada iteração ele testa se há uma transicao válida. Se não, ele interrompe o *while*. Após isso exibe a mensagem se a palavra é valida ou não, se é aceita ou não.

```

def automato():
    k = 0
    estado = 0
    palavra = arq[0]
    while k < len(palavra):
        est=estado
        if palavra[k] == '0':
            estado = defineTransicao(transicao[estado][1][1:])
        else:
            estado = defineTransicao(transicao[estado][2][1:])
        if estado == -1:
            break
        print("Delta(q{},{})=q{}".format(est,palavra[k],estado))
        k+=1
    if estado == int(arq[2][1:]):
        print("A cadeia foi aceita")
    else:
        print("A cadeia foi rejeitada")

```

E por ultimo temos a execução das funções.

```
leArquivo(argv[1])
```

```
separaArquivo()
```

```
automato()
```